# An Integral Simplex Method for Solving Combinatorial Optimization Problems

by

Gerald L. Thompson*

Carnegie Mellon University
PITTSBURGH, PENNSYLVANIA 15213

Graduate School of Industrial Administration
WILLIAM LARIMER MELLON, FOUNDER

Management Science Research Report #MSRR-620

# An Integral Simplex Method for Solving Combinatorial Optimization Problems

by

Gerald L. Thompson*

July 1996

Management Science Research Group
Graduate School of Industrial Administration
Carnegie Mellon University
Pittsburgh, PA 15213

*Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA 15213. Research supported by Office of Naval Research

# An Integral Simplex Method for Solving Combinatorial Optimization Problems

by

Gerald L. Thompson

Carnegie Mellon University, Pittsburgh, PA 15213

July 11, 1996

### Abstract

In this paper a local integral simplex method will be described which, starting with the initial tableau of a set partitioning problem, makes pivots using the pivot on one rule until no more such pivots are possible because a local optimum has been found. If the local optimum is also a global optimum the process stops. Otherwise, a global integral simplex method creates and solves a search tree consisting of a polynomial number of subproblems, subproblems of subproblems, etc., and the solution to at least one of which is guaranteed to be an optimal solution to the original problem. If that solution has a bounded objective then it is the optimal set partitioning solution of the original problem, but if it has an unbounded objective then the original problem has no feasible solution. It will be shown that the total number of pivots required for the global integral simplex method to solve a set partitioning problem having $m$ rows, where $m$ is an arbitrary but fixed positive integer, is bounded by a polynomial function of $n$. Preliminary computational experience is given which indicates that global method has a low order polynomial empirical performance when solving such problems.

## 1. Introduction

In this paper a new approach to solving some kinds of zero/one integer programming problems will be presented which (a) is independent of previous popular approaches such as branch and bound or cutting plane methods; (b) depends only on linear algebra and linear programming; (c) requires only a polynomial number of pivots to solve any set partitioning problem having $m$ rows, where $m$ is an arbitrary but fixed positive integer; and (d) provides a certificate of optimality for the final solution. The complete theoretical development of the method and preliminary computational results are presented.

The basic idea of the method of this paper is to use a modified version of the simplex method, called the *local integral simplex method,* in which pivot steps are made only on ones in the simplex tableau. Since the problems considered here are set partitioning problems, the initial pivots on ones are easy to find. However, in later steps the integral simplex method frequently stops at a point at which there are columns having negative indicators which could improve the objective function if brought into the basis, but the minimum ratio rule can find pivots only on numbers larger than one. In other words the method finds only a local optimum for the problem. In order to find the overall optimum another method ,called the *global integral simplex method*, is devised which uses the local simplex method as a subroutine. Once the local method stops at a local optimum the global method defines a subproblem for each negative reduced cost in the local method's final tableau. It is shown in this paper that the best solution found among all of the subproblems is the global optimum (if there is one) for the original problem. It is also shown that the number of pivot steps needed for the global integral simplex method to solve a set partitioning problem having $m$ (where $m$ is fixed) rows is bounded by a polynomial function of $n$.

In previous papers Balas and Padberg [1972, 1975] have shown that the simplex method, when restricted to pivoting only on ones can be used to explore the integer solutions within the feasible set of the linear programming relaxation of the set partitioning problem. In their 1975 paper they provide two algorithms for solving set partitioning problems with their pivoting method together with column generation techniques.

## 2. The Local Integral Simplex Method (LISM)

Many combinatorial optimization problems have the following form:

$$\begin{aligned} &\text{Minimize} \quad cx \\ &\text{Subject to} \ \ Ax <=, =, >= \ b \\ &\qquad\qquad\qquad x \ \text{ is a 0/1 vector} \end{aligned}$$

where $A$ is an $mxn$ matrix with 0/1 entries, $x$ is an $nx1$ column vector, $b$ is an $mx1$ vector with positive integral entries, and $c$ is a $1xn$ vector with positive integral entries.

For instance the *set partitioning problem* can be stated as:

$$
\begin{array}{lll}
\text{Minimize} & cx + My & \\
\text{Subject to} & Ax + \;\; y = f & \qquad\qquad (1) \\
 & \quad x \text{ is a 0/1 vector} & \\
 & \quad y >= 0 &
\end{array}
$$

where $A$ is an *mxn* 0/1 matrix, $x$ is an *nx*1 vector, $y$ is an *mx*1 vector whose components are called *artificial variables*, $c$ is an 1*xn* vector of integer costs, $f$ is an *mx*1 vector of all ones and $M$ is a large number. All the example problems considered in this paper will be set partitioning problems. Solutions $(x, y)$ to this problem have coefficients 0 or 1, and are called *partition solutions.* These are the only solutions calculated by the integral simplex method. A partition solution $(x, y)$ is said to be an *integral partition* if $y = 0$.

In the LP relaxation of the problem, the condition "$x$ is a 0/1 vector" in (1) is replaced by the condition "$x >= 0$". The partition solutions just defined are extreme points of $X$, the set of all feasible solutions to the LP relaxation. Moreover, $X$ has other extreme points, called *fractional partitions, (x, y)*, whose components are real numbers in the closed interval [0,1].

We use the following notation for the tableaus of the problem. Let $T$ be the $(m+1)x(n+m+1)$ initial tableau of a set partitioning problem in which the *mxn* matrix $A$ is put in rows 1,...,$m$ and columns 1,...$n$, the artificial vectors are put in rows 1,...,$m$ and columns $n$+1,...,$n$+$m$, the $b$ vector is put in column 0, the $c$ vector is put in row 0 in columns 1,...,$n$, and the costs $M$ of the artificial variables are put in row zero and columns $n$+1,...,$n$+$m$. In row 0 and column 0 a zero entry is placed which will record the values of the objective function. In the LP phase I step the artificial vectors are pivoted in, to become the initial basis, and the last $m$ columns are dropped to give a compact initial tableau. The entries in row 0, columns 1,...,$n$ are called *indicators* of the corresponding variables that label the columns, instead of reduced costs as in ordinary LP, because there are no dual variable prices. The indicators of the basic variables whose labels appear on the righthand side of the tableau are all *zeros.*

The steps of the local integral simplex method, written in a psuedo C language, are shown in Figure 1. In that figure there are two places where a RULE is referred to. By that is meant one of the following rules: Bland's rule for preventing cycling, the lexicographic ordering rule for preventing cycling, or random selection rule for preventing cycling with probability one.

```
SET UP:
    Create the initial tableau of the problem and perform phase I as described
    Let $T$ be the resulting tableau with basic and non basic variables labeled
PIVOT STEP:
CHOOSE PIVOT{
    Make a list $L$ of columns of $T$ having negative indicators
    If ($L$ is not empty){
        Apply RULE to select and remove an element $j$ of $L$
        Find  minratio = minimum $t_{i0}/t_{ij}$
                                 $t_{ij} > 0$
        Make a list $P$ of rows $i$ with $t_{ij}$=1 and $t_{i0}$=minratio
        If ($P$ is not empty){
            Use RULE to select and remove an element of $P$
            Go to UPDATE
        }
    }
    goto HALT
}
UPDATE{
    Solve the problem using the usual updating steps of the simplex method
    Go to PIVOT STEP
}
HALT: /* A locally optimal solution has been found */
```

Figure 1.  Psuedo C code for LISM.

When LISM stops its final tableau may have all nonnegative indicators which, as we shall see, means that a global optimum has been found. If LISM has some negative indicators, then it is possible that there are some better integral solutions. The global integral simplex method will find such better solutions. To do so it defines subproblems by fixing to one each variable that has a negative indicator, and sets up and solves each of these subproblems (which in turn may require solving subproblems of subproblems, etc.) so defined.

## 3. The Global Integral Simplex Method (GLISM)

We begin this section by making some important definitions.

*Definition* 1. Two zero/one column vectors $a$ and $b$ with the same number of components are said to be *consistent* if their inner product $a \cdot b$ is zero, i.e., if for every row $i$ the product $a_i b_i = 0$. They are said to be *inconsistent* if for some row $i$ the product $a_i b_i = 1$.

*Definition* 2. Let $T$ be the tableau of the original problem, let $T_j$ be the jth column of $T$ and let $j^{(1)}, \ldots, j^{(k)}$ be a set of indices of a consistent set of columns of $T$; then the subproblem defined by setting $x_j^{(1)} = 1, \ldots, x_j^{(k)} = 1$ is obtained by doing the following steps: (a) let $w = T_j^{(1)} + \ldots + T_j^{(k)}$; (b) cross out inconsistent columns $j$ of $T$ for which $w \cdot T_j > 0$; (c) cross out rows $i$ of $T$ such that $w_i = 1$; and (d) eliminate columns of $T$ that have all zeros in the remaining rows.

*Definition* 3. Let $T$ be the final tableau of a set partitioning problem or subproblem, and let $x$ be its partition solution shown in $T$ for the problem. If $x^*$ is a feasible partition solution, $x_j^* = 1$, and $x_j$ is a nonbasic variable labeling a column of $T$ having a negative indicator, then $x_j$ is said to be an *integral set partition flag variable* for $x^*$.

*Definition* 4. Let $T$ be the final tableau of a set partitioning problem or subproblem, and let $x$ be its partition solution shown in $T$ for the problem. If $x^*$ is a feasible fractional partition solution , $x_j^*$ belongs to the open interval (0,1), and $x_j$ is a nonbasic variable labeling a column of $T$ having a negative indicator, then $x_j$ is said to be a *fractional set partition flag variable* for $x^*$.

In case it is not known whether a nonbasic column with negative indicator is an integral flag variable or a fractional flag variable, which is most of the time, it will simply be referred to as a *flag variable*. As we shall see, the same variable may be a flag variable for one or more integral set partition solutions and simultaneously be a flag variable for one or more fractional set partition solutions.

The global integral simplex method GLISM constructs a search tree of subproblems which are solved using LISM as a subroutine. It will be shown that the optimal solution to the subproblem which has the smallest objective function value is also an optimal solution to the original problem, provided it has a solution. The psuedo C code for the global simplex method is given in Figure 2.

```
BEGIN:
        Make a copy $T$ of the tableau of the original problem
START{/* Create the tableau of problem $P$ which has variables $x_j^{(1)},\ldots,x_j^{(k)}$ set to one*/
        Set the vector $w = T_j^{(1)} + T_j^{(2)} + \ldots + T_j^{(k)}$
        Cross out all inconsistent columns $j$ of $T$ for which the inner product $w \cdot T_j > 0$
        Cross out all rows $h$ of $T$ that have $w_h=1$
        Eliminate any zero columns in $T$
}
SOLVE{
        Using LISM, solve the problem in $T$, saving its solution if it is the best so far
        Add to the list $Q$ all of the new subproblems created by the negative indicators in
                the final tableau $T^f$
}
SELECT{
        If ($Q$ is empty) goto HALT
        Else choose and remove a problem $P$ from $Q$
        Go to BEGIN
}
HALT: /* The best solution found is optimal */
```

Figure 2. Psuedo C code for GLISM

## 4. Theoretical Results

Here we will derive all of the theoretical results that are needed to prove that the LISM and GLISM algorithms will find optimal solutions to set partitioning problems. Although some of the results are similar to those that are true for the theory of linear programming, others are not and the reader should be careful to follow the interpretations and arguments as presented.

*Property 1.* Let $T$ be the tableau of a locally optimal integer partition solution $x$.

(a) The objective value of $x$ is the negative of the entry in the first row and first column of $T$.

(b) If $x^*$ is any other locally optimal partition then its objective value is the objective value of $x$ plus the sum of the indicators of $T$ corresponding to the components of $x^*$ that are equal to 1.

*Proof.* Property 1(a) follows because the problem being solved by the simplex method is to maximize the negative of the objective function. Property 1(b) follows directly from the *invariance of linear relations property* which is: *Any linear relation that holds for vectors in one basic solution holds for the same vectors in every other basic solution.* The latter holds because linear transformations, such as the pivot transformations of the simplex method, preserve linear relations.

*Theorem 1.* Let $T$ be the tableau of a locally optimal partition $x$ which has some negative indicators and assume there is a better locally optimal integer partition $x^*$. Then

(a) $T$ contains a column labeled with a nonbasic variable $x_j$ having a negative indicator, and such that $x_j^* = 1$; i.e., $x_j$ is a set partition flag variable for $x^*$.

(b) $x^*$ is a locally optimal partition for the subproblem of the current problem which is defined by setting $x_j$ equal to one.

*Proof.* (a) Let $x$ and $x^*$ be partition solutions having the properties stated in the hypothesis of the theorem. By Property 1(b) the objective function of $x^*$ is equal to the sum of the the objective function of $x$ plus the indicators of components of $x^*$ for which $x_j^*$=1. Since $x^*$ has a smaller objective value that $x$ does, at least one of the indicators of $x^*$ for which $x_j^*$=1 must be negative. Since the indicators of the basic variables are always equal to 0, it follows that $x_j^*$ must be nonbasic.

(b) From the assumptions, $x_j{}^*= 1$ in $x$*. Also the column labeled $x_j$ is consistent with the column of every other variable $x_k^*$ that equals 1 in $x$*, so that the subproblem defined by $x_j = 1$ will contain all the variables that are equal to one in $x^*$, except for $x_j$ itself. Since $x^*$ is optimal in the original problem it will also be optimal in the new subproblem. For suppose on the contrary, that $x^{**}$ is better than $x^*$ in the subproblem. Then all column vectors $h$, with $x_h{}^{**}$ =1 in the subproblem must be consistent with $x_j^*$, which means that $x^{**}$ must have been better than $x^*$ in the original problem, a contradiction.

*Corollary.* Either the current solution is globally optimal for the current problem, or else $T$ will have a column whose label is a set partition flag variable for such a globally optimal solution $x^o$.

The result in the corollary is much stronger than any similar result for linear programming. In the simplex solution of an LP problem it rarely happens that a variable having a negative indicator at some point during solution process will belong to a globally optimal solution. But in the case of LISM it happens for at least one variable in every problem having negative indicators.

*Theorem 2*. Let $T$ be the tableau of a locally optimal partition $x$ which has some negative indicators and assume there is a better locally optimal fractional partition $x^*$. Then

(a) Tableau $T$ contains a column whose label $x_j$ is a fractional set partition flag variable for $x^*$.

(b) If $x_j$ is an integral variable of $x^*$, i.e., $x_j^* = 1$, then $x^*$ is a locally optimal fractional partition for the subproblem that is defined by setting $x_j = 1$.

(c) Let $x_j^* < 1$ be a fractional variable of $x^*$; then $x^*$ is *not* a locally optimal partition for the subproblem that is defined by setting $x_j = 1$.

*Proof:* The proofs of (a) and (b) are similar to the proofs of the previous theorem.

To prove part (c), recall that if $x_j^*$ labels column $j$ and is a fractional variable then there is at least one other column which is labeled by a fractional variable $x_k^*$ and for these two columns there is at least one row in which they both have a 1, i.e., they are inconsistent. Hence the subproblem $P$ defined by setting $x_j^*$ equal to 1 does not contain the column labeled by $x_k^*$ which means that the fractional solution $x^*$ is not feasible for problem *P*.

Property (c) of this theorem is the key result that shows how the global simplex method eliminates fractional partitions as candidates for globally optimal integer partitions by defining subproblems as the search tree is being created. Notice that when the local integral simplex method stops at a local minimum the global integral simplex method merely searches for negative indicators in the tableau and sets the corresponding variable equal to one to create a new branch in the search tree without trying to find out whether that variable corresponds to (a) an integral set partition, or merely to (b) a fractional set partition. In the case (a) the tree may branch further to locate other, and perhaps better integral solutions, while in case (b) the fractional partition solution simply vanishes, and does not appear anywhere lower in the subproblem search tree.

*Theorem 3*. Assume that the final tableau $T$ of a problem (or subproblem) has all nonnegative indicators.

(a) If all artificial variables have value zero in the final tableau, the solution displayed in that tableau is an optimal set partition for the problem.

(b) If one or more artificial variables have value one in the final tableau, the problem has no feasible set partition.

*Proof.* (a) Under the stated assumption, by Property 1(b) every other feasible partition has a cost greater than or equal to that of the solution shown in $T$. Since the displayed solution is feasible it therefore is globally optimal.

(b) Under the stated assumptions it follows from Property 1(b) that the objective function is greater or equal to $M$, and by Property 1(a), all other solutions have objective value greater than or equal to $M$. Since $M$ can be arbitrarily large, and since feasible solutions have finite values, it follows that there is no feasible partition.

*Theorem 4.* For any set partitioning problem the global integral simplex method will either find a globally optimal integral set partition, or else prove that no such feasible partition exists.

*Proof.* The convergence of LISM when used to solve subproblems can be proved as in the simplex method by using the well known properties of the RULE in Figure 1. By Theorem 2, either the set partition which appears in the final tableau of the first subproblem solved, or one of the set partitions whose flag variable appears in the tableau at a local optimum of one of the subproblems, is a globally optimal solution. Since the global integral simplex method finds an optimal solution for every subproblem corresponding to negative indicators in the final tableau, the subproblem yielding the best locally optimal solution is therefore globally optimal. If that solution does not contain any artificial variable with value one then it is a globally optimal set partition. If that solution does contain one or more artificial variables with value one, then by Theorem 3(b) the original problem does not have any feasible partition.

## 5. Computational Complexity of the Algorithms

In this section it is shown that both LISM and GLISM have polynomial complexity when used to solve set partitioning problems having a fixed number $m$ of rows, by showing that the maximum number of pivots required to solve either of them is bounded by polynomial function of $n$.

*Theorem 5.* The maximum number of pivots needed by LISM to find a locally optimal solution to a set partitioning problem having $m$ rows and $n$ columns is at most $(n+1)^m$.

*Proof.* At each pivot LISM moves from one feasible basis having $m$ columns to another. Let $S$ be the set, having $m+n$ elements, that consists of the $m$ artificial column vectors together with the $n$ column vectors of $A$. Then the maximum number of pivots required for LISM to converge to a local optimum is less than or equal to the maximum number of $m$ element subsets of $S$, which is the combinatorial coefficient

$$\binom{n+m}{m} = (n+m)!/(m!\,n!) = (n+1)\cdot(n+2)\cdots(n+m)/m!$$
$$< (n+1)\cdot((n/2)+1)\cdots((n/m)+1)$$
$$< (n+1)^m \;.$$

which gives the upper bound asserted in the theorem.

*REMARK.* In spite of the Klee-Minty (KM) examples, the same argument shows that the same bound also holds for the ordinary simplex method using any incoming vector rule, when solving a linear program having $m$ rows and $n$ columns. The reason is that the KM examples are square, so that the largest KM example having $m$ rows also has $m$ columns which implies that its KM bound is $2^{m-1}$ which is less than the bound stated in the theorem. It is also easy to show that the KM examples can be solved in *one* pivot when using either the least negative (instead of the most negative) reduced cost rule, or the maximum z-change rule, which means that the KM examples are of little importance in complexity theory.

*Theorem 6.* If $n >= 2$ the total number of pivots required by the GLISM to solve an $mxn$ set partitioning problem is less than $2(n^2+n)^m$.

*Proof.* The original problem at level 0 of the subproblem search tree has at most $n$ negative indicators, and hence it creates at most $n$ subproblems at level 1 of the search tree. Each of these subproblems is of size at most $(m\text{-}1)x(n\text{-}1)$, and hence creates at most $n(n\text{-}1) < n^2$ subproblems. Similarly, the size of subproblems on the kth level of the tree is at most $(m\text{-}k)x(n\text{-}k)$ and hence they create at most $n(n\text{-}1)\ldots(n\text{-}k) < n^k$ subproblems. The deepest level of the tree is $(m\text{-}1)$, since "subproblems" on the level $m$ would have no rows. Hence the number of subproblems on the deepest possible level is at most $n(n\text{-}1)\ldots(n\text{-}m+1) < n^m$. Adding these together, and using the fact that $n >= 2$ implies $n/(n\text{-}1) <= 2$, gives the total number of subproblems that must be solved as

$$n + n^2 + \ldots + n^m = n(n^m \text{-}1)/(n\text{-}1) < 2n^m.$$

Multiplying this result by the result of Theorem 5 gives the bound stated in the theorem.

In the theory of computational complexity, see Garey and Johnson [3], $P$ is the class of problems that can be solved in a number of steps which has a polynomial bound expressed in terms of the basic input data. The class $NP$ consists of the problems which can be solved in a number of steps which has an exponential bound. Among the latter is the subclass of problems, called $NP$ complete problems, which have the property that, if any member of the class can be solved in a number of steps having a polynomial bound then so can all members of the class. In fact, there is a mapping between a polynomial algorithm for any member of this class to any other member. It is well known that set partitioning problems are in the $NP$ complete class. It follows that any member of the $NP$ complete class having a 0/1 matrix structure similar to the set partitioning problem, can be solved by some variant of the algorithms discussed in this paper.

## 6. A Small Example

Figure 1 contains the data for a 5 x 11 numerical example.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 72 | 48 | 77 | 44 | 56 | 49 | 77 | 41 | 47 | 96 | 42 | |

Figure 1. Data for a 5x11 example.

In order to solve the problem using LISM, five artificial vectors are added, and a phase I start is made to get an initial feasible basis. After pivoting seven times and eliminating the artificial vectors the final compact tableau shown in Figure 2 is obtained. The set partitioning solution shown in Figure 2 is $x_1$=$x_7$=1 with objective value $z = 149$.

$$\begin{array}{cccccc|c|l}
x_5 & x_3 & x_4 & x_9 & x_{10} & x_2 & & \\
\hline
1 & 1 & 1 & 0 & 0 & 0 & 1 & x_7 \\
0 & -1 & -1 & 1 & 1 & 0 & 0 & x_{11} \\
0 & 0 & -1 & 0 & 1 & 0 & 0 & x_8 \\
-1 & -1 & -1 & 1 & 1 & 1 & 0 & x_6 \\
2 & 2 & 2 & -1 & -1 & -1 & 1 & x_1 \\
\hline
-116 & -53 & -45 & 28 & 36 & 71 & -149 &
\end{array}$$

Figure 2. Local optimum found by LISM.

We see from the figure that there are three flag variables, $x_3$, $x_4$, and $x_5$ each of which defines a subproblem, as follows:

(i) Setting $x_3$=1 gives the 2x3 subproblem with rows 1 and 2, and columns 2, 9, and 11 in Figure 11. The optimal integral solution to this subproblem (together with $x_3$=1) is: $x_3$=$x_9$=1 and $z$= 124.

(ii) Setting $x_4$=1 gives the 3x5 problem with rows 1, 2, and 3, and columns 2, 8, 9, 10, and 11. The optimal integral solution to this subproblem (together with $x_4$=1) is: $x_4$=$x_8$=$x_9$=1 and $z$= 132. There is another nonoptimal solution to this problem, namely, $x_4$=$x_{10}$=1 and $z$= 140 which was not found by LISM since it found the best one first. Both of these solutions had $x_4$ as an integral set partition flag variable.

(iii) Setting $x_5$=1 gives a 1x1 problem with row 1 and column 2. The optimal integral solution to this subproblem (together with $x_5$=1) is: $x_2$=$x_5$=1 and $z$= 104.

Since all three subproblems yielded optimal integral solutions there were no more subproblems to be solved, and the best of the solutions, given in (iii), is optimal for the problem in Figure 1.

## 7. Certificate of Optimality

In ordinary linear programming theory the optimal dual solution gives a *certificate of optimality* in the form of a bounding hyperplane $P$, the direction of whose normal is given by the optimal dual solution. This hyperplane contains at least one point of the convex set of feasible primal solutions $X$, namely, the optimal solution found by solving the linear programming problem, and all of the other feasible solutions in $X$ lie either on the same side of $P$ or actually on the hyperplane $P$.

It is possible to find something similar here, but not by using a dual solution to the problem, because there is usually no feasible dual problem in the case of integer programming. Suppose that we somehow knew of the solution $x_2$=$x_5$=1 and $z$= 104 and wanted to see if it were optimal. We start with the original problem in Figure 1, add the artificial vectors and perform the Phase I step; then we pivot to bring into the basis the columns of the known solution, namely, the columns labeled $x_2$ and $x_5$, After that is necessary to pivot several more times to arrive at the locally optimal solution shown in the full tableau of Figure 3.

Note that in that figure the optimum solution is as found in (iii) above; however there is a negative indicator with label $x_6$. As usual we set $x_6$=1 and find that the corresponding subproblem has no columns since $x_6$ is not consistent with any other column in Figure 1. Hence this subproblem does not have any finite integral solution which provides a second proof that the solution $x_2$=$x_5$=1 and $z$= 104 is optimal. (The first proof was given by finding the solutions to the three subproblems in (i), (ii) and (iii) above.)

This is the certificate of optimality for the problem at hand. In the general case the procedure for finding the certificate of optimality is the same once an optimal solution is found by the GLISM. In order to find a certificate of optimality, set up the original tableau, and pivot on ones until the variables in the optimal solution become basic, then continue pivoting until a local optimum solution for the tableau is obtained. Then show that, for each negative indicator in the final tableau, the corresponding subproblems do not have better integral solutions. Note that here the certificate of optimality is not obtained as an automatic by product of finding an optimal solution to a linear program. It requires additional optional computations which should be done only if it is felt that the results might be useful, for instance, in interpreting the optimal solution.

In order to find the solution that corresponds to the flag variable $x_6$ in Figure 3 we temporarily drop the pivot on one rule in order to return to ordinary linear programming, and pivot on the 2 entry in row 4 and column 6 in Figure 3. We obtain the optimal *fractional* solution $x_5$=$x_6$=$x_7$=1/2 and $z$= 91. This is, in fact, the LP relaxation solution of the problem shown in Figure 1. What

| $x_1$ | $x_3$ | $x_4$ | $x_{11}$ | $x_{10}$ | $x_6$ | | |
|---|---|---|---|---|---|---|---|
| -1 | 0 | 0 | 0 | 0 | -1 | 0 | $x_7$ |
| 0 | -1 | -1 | 1 | 1 | 0 | 0 | $x_9$ |
| 0 | 0 | -1 | 0 | 1 | 0 | 0 | $x_8$ |
| 1 | 1 | 1 | -1 | 0 | 2 | 1 | $x_2$ |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | $x_5$ |
| 45 | 20 | 28 | 43 | 8 | -26 | -104 | |

Figure 3. Final tableau after pivoting in the columns labeled $x_2$ and $x_5$ and reoptimizing.

this means is that the flag variable $x_5$ back in Figure 2 was both an integral flag variable for the optimal solution found in (iii), and also a fractional flag variable for the LP relaxation solution just found. When $x_5$ became basic with value one in Figure 3 it could no longer be a flag variable for the fractional solution of the LP relaxation, so that role was taken over by $x_6$.

## 8. Computational Experience

Some limited experience has been gained with the program. The data from a series of 25 row problems is shown in graph of Figure 3. Note that in the figure, there are also the results from a

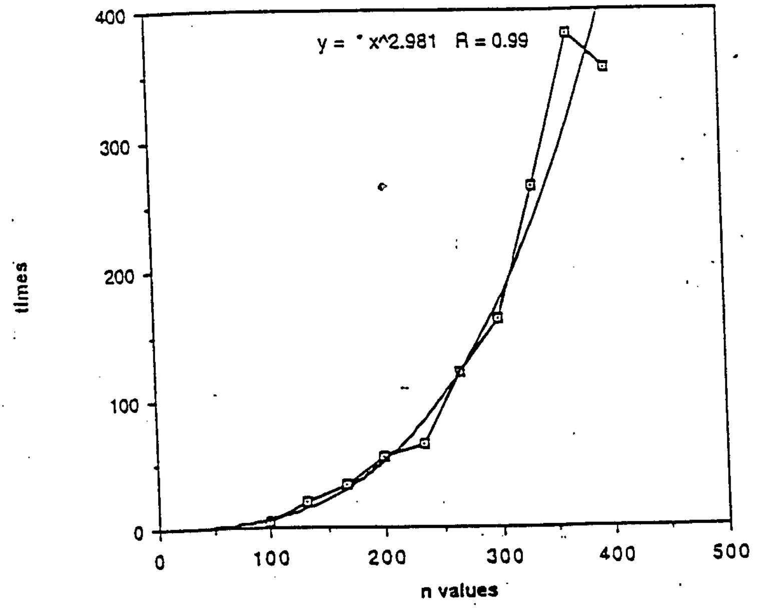

Figure 4. Plot of solution times for 25 row problems.

logarithmic fit of the data that show that the closest fit to the solution time versus number of columns is approximately a cubic function.

Additional computational testing using both sequential and parallel computers is under way for these methods, which will be reported on in the near future.

It is well known that empirically the computational difficulty of solving a linear program having a fixed number of rows *m*, is linear in *m*, e.g. approximately 2*m* or 3*m*, and that is what makes the simplex method be so successful in solving practical problems. If it turns out that the empirical difficulty of solving combinatorial optimization problems with the algorithms in this paper is always a low degree polynomial, then these algorithms may be equally successful in solving real applied problems.

## 9. Conclusions

In this paper two simple algorithms, which are based on adding the pivot on one rule to the simplex method together with the creation of a search tree of subproblems, has resulted in a zero/one integer programming method that requires only a polynomial number of pivots either, to find an optimal solution to a given set partitioning problem, or else, to prove that no feasible partition exists for the problem. It was also shown emprically that the method is likely to be successful in practice and offers hope of being a useful tool for solving a large variety of practical problems.

## Bibliography

[1] Balas, E., and M. Padberg, (1972), "On the Set Covering Problem," *Operations Research,* 20, 1153-1161.

[2] Balas, E., and M. Padberg, (1975), "On the Set Covering Problem: II. An Algorithm for Set Partitioning, " *Operations Research,* 23, 74-90.

[3] Garey, Michel R. and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* Freeman Press, 1979.